



Politechnika Wrocławska

---

Wydział Elektroniki  
Katedra Informatyki Technicznej

PROJEKTOWANIE EFEKTYWNYCH ALGORYTMÓW  
PROJEKT

---

**Projekt nr 2 - Implementacja i analiza  
efektywności algorytmu Tabu Search i/lub  
Symulowanego Wyżarzania problemu  
komiwojażera.**

---

Miron Oskroba, 236705

Prowadzący - dr inż. Jarosław Mierzwa  
Termin zajęć: Pn 15<sup>15</sup>

Wrocław, 21.12.2020

# Spis treści

<b>1</b>	<b>Wstęp teoretyczny</b>	<b>2</b>
<b>2</b>	<b>Opis najważniejszych klas</b>	<b>3</b>
<b>3</b>	<b>Plan eksperymentu</b>	<b>4</b>
<b>4</b>	<b>Wyniki eksperymentów</b>	<b>5</b>
4.1	Tabele . . . . .	5
4.2	Wykres . . . . .	8
4.3	Najlepsze rozwiązania . . . . .	9
4.3.1	ftv47.atsp . . . . .	9
4.3.2	ftv170.atsp . . . . .	10
4.3.3	rbg403.atsp . . . . .	10
<b>5</b>	<b>Wnioski dotyczące otrzymanych wyników</b>	<b>11</b>
<b>6</b>	<b>Wnioski</b>	<b>11</b>
<b>7</b>	<b>Literatura</b>	<b>11</b>

# 1 Wstęp teoretyczny

Celem wykonywanego zadania jest implementacja i analiza efektywności Tabu Search i/lub Symulowanego Wyżarzania dla asymetrycznego problemu podróżyującego komiwojażera. Zaimplementowane algorytmy:

- Algorytm SW dla ATSP

Algorytm Symulowanego Wyżarzania (z ang. 'simulated annealing') to algorytm rodzaju heurystycznego (bez gwarancji znalezienia rozwiązania optymalnego [1]), ciągle przeszukujący przestrzeń alternatywnych rozwiązań do momentu spełnienia warunku stopu. Algorytm posiada elementy:

- Temperatura - Początkowo ustalana w zależności od postaci rozpatrywanej instancji, rozpatrywana jest w sposób następujący. Należy porównywać koszty losowo wygenerowanych tras i wydobyć największą różnicę kosztów. W głównej pętli algorytmu (pętla, w której jedynie warunek stopu w sekundach przerwie algorytm) należy aktualizować temperaturę (ochładzać przemnażając przez odpowiednio dobrany *wsp. mocy chłodzenia*). W tej wersji algorytmu zaimplementowane również zostało ogrzewanie. W momencie kiedy algorytm zakończyłby pracę po osiągnięciu temperatury minimalnej, zamiast tego, jednorazowo ustawia temperaturę na wyższą w zależności od aktualnego najlepszego rozwiązania oraz rozmiaru badanej instancji.

- Moc chłodzenia - Współczynnik Mocy, z którą wykonywane jest chłodzenie. Wg wzoru

$$Temp = temp\_start / (cool\_rate * it)$$

- Temperatura minimalna - Temperatura, po przekroczeniu której nastąpi grzanie.
- Warunek stopu - Czas w sekundach, po którym algorytm przerwie działanie i zwróci wyniki.
- Macierz Kosztów - wczytana z pliku, w tym przypadku będzie to jeden z trzech plików: 'ftv47.atsp', 'ftv170.atsp' lub 'rbg403.atsp'.

Sposób wyboru rozwiązań sąsiednich został zrealizowany w sposób następujący. Na podstawie losowo wygenerowanych indeksów  $i, j$  będących w zakresie instancji, należy porównać trzy nowo powstałe [2], tymczasowe listy pod kątem kosztu trasy, stosując:

- inwersję (z ang. 'inverse')
- wstawianie (z ang. 'insert')
- zamianę (z ang. 'swap')

Po znalezieniu najmniejszego kosztu z wypisanych możliwości, zostaje nadpisane aktualne rozwiązanie. Porównywane jest rozwiązanie aktualne z rozwiązaniem najlepszym. Następnie sprawdzane jest czy rozwiązanie sąsiednie jest lepsze od aktualnego, które zostanie zaakceptowane, jeśli:

- koszt rozwiązania sąsiedniego jest niższy niż aktualnego lub
- wylosowana liczba typu *double* z przedziału  $<0,1>$  będzie mniejsza od wyrażenia

$$\exp(-(cost\_neighbor - cost\_current)/temp)$$

na podstawie[2].

## 2 Opis najważniejszych klas

Projekt został napisany w IDE IntelliJ2020.3 CLion w języku C++. Najważniejsze klasy prezentują się następująco (Opis funkcji jest poglądowy, bez zwracanych wartości):

- **main** - Punkt wejściowy programu. Zawiera interfejs użytkownika:
  1. Wczytanie danych z pliku
  2. 2.zmien maksymalny czas wykonywania[s]
  3. 3.zmien wsp. zmiany temperatury
  4. 4.SW-all -Wykonaj testy algorytmu SW dla podanych ustalonych parametrów dla plików *ftv47.atsp*, *ftv170.atsp*, *rbg403.atsp*. Każdy plik jest testowany 10 razy, czas wykonywania ustawiony odpowiednio: 2,4,6 [min].
  5. 5.SW-single (Wykonaj pojedynczy test)
  6. 0.Wyjście
- **SW** - Klasa *Symulowanego Wyżarzania*. Aby uruchomić algorytm należy najpierw zainicjować instancję klasy podając w konstruktorze niezbędne parametry:
  - macierz kosztów
  - warunek stopu w sekundach
  - moc chłodzenia

Funkcje w klasie:

- **find\_best\_path()** - główne ciało algorytmu. W pierwszej kolejności przygotowuje niezbędne zmienne i dane do prawidłowej pracy nad aktualnie wczytaną instancją z pliku, następnie algorytm trwa do określonego momentu - do momentu warunku stopu wyrażonego w [s] przez użytkownika. W tej wersji algorytmu zaimplementowane zostało ogrzewanie - po przekroczeniu temperatury minimalnej jednorazowo ustawiana jest aktualna temperatura, celem nieprzerwanej pracy przez określony czas i polepszenia najlepszego rozwiązania.
- **initial\_temperature()** - liczy temperaturę początkową dla aktualnej postaci macierzy kosztów. Należy porównywać koszty losowo wygenerowanych tras i wydobyć największą różnicę kosztów [3].

```
1      double SW::initial_temperature()
2      {
3          double initial_temp = 0.;
4          double tmp_diff;
5
6          int dim = cost_matrix.size() - 1;
7          vector<int> permutation_first(dim);
8          vector<int> permutation_second(dim);
9          iota(permutation_first.begin(), permutation_first.end(), 1);
10         permutation_second = permutation_first;
11
12         for (int i = 0; i < 1000; i++)
13         {
14             randomize(permutation_first);
15             randomize(permutation_second);
16             tmp_diff = abs(cost(permutation_first) - cost(permutation_second));
17             if(tmp_diff > initial_temp)
18                 initial_temp = tmp_diff;
19         }
20         return initial_temp;
21     }
```

- **cost(vector<int> path)** - funkcja liczenia kosztu danej ścieżki.
- **swap\_greedy(vector<int> &path)** - schemat szukania rozwiązania sąsiedniego dla ścieżki *path*, opisany w 3.1 w [2]. Wybiera jednego z trzech najlepszych kandydatów na bycie rozwiązaniem sąsiednim - pod względem kosztu przebycia danej drogi, minimalny staje się rozwiązaniem sąsiednim, które jest porównywane w dalszej części algorytmu z: rozwiązaniem najlepszym (globalnym) oraz lokalnym (aktualnym). Bardzo istotnym elementem w tej części algorytmu tzn. wybierania najlepszego sąsiada jest czas wykonywania operacji swap, inverse i insert.
- **\_\_swap(vector<int> path, vector<int>& path\_dest, int i, int j)** - Zamienia podane indeksy wektora.
- **\_\_inverse(vector<int> path, vector<int>& path\_dest, int i, int j)** - Dokonuje inwersji na indeksach i ,j.
- **\_\_insert(vector<int> path, vector<int>& path\_dest, int i, int j)** - Dokonuje wstawienia elementu j przed element i;
- **randomize(vector<int>& path)** - 'mieszanie' - zmienianie ścieżki *path* określoną ilość razy tak, aby można było ją uznać za względnie losową.
- **Timer** - Umożliwia niezależne odmierzenie czasu (warunki stopu, czas wykrycia rozwiązania najlepszego)
  - **start()**
  - **stop()**
  - **info()**
  - **stop\_test()**
- **FileHandler** - Umożliwia wczytywanie plików .atsp do programu. Umożliwia także podstawowe debugowanie wczytywanych plików.
  - **read\_from\_file(const char\* datafile);**
  - **print\_matrix(std::vector<std::vector<int>> matrix);**

### 3 Plan eksperymentu

Plan eksperymentu odbędzie się zgodnie z wytycznymi projektowymi; dla plików zostanie wykonana następująca testy:

Tablica 1: Plan eksperymentu.

nazwa pliku	ilość testów	czas testu[min]	wartość oczekiwana
ftv47.atsp	10	2	1776
ftv170.atsp	10	4	2755
rbg403.atsp	10	6	2465

Dla otrzymanych wyników policzone zostaną błędy względne wg wzoru:

$$|f_{zn} - f_{opt}| / f_{opt} * 100\%, \text{ gdzie}$$

$$f_{zn} - \text{warto obliczona przez algorytm}$$

$$f_{opt} - \text{warto oczekiwana (optymalna)} - \text{najlepsze znane rozwiązanie}$$

Dodatkowo, dla każdego uruchomienia wypisane zostaną:

- najlepszy wynik danego oraz uruchomienia
- jego czas wystąpienia .

## 4 Wyniki eksperymentów

### 4.1 Tabele

Wyniki eksperymentów dla współczynników opisanych w Plan eksperymentu zaprezentowane są w tabelach:

- 2
- 3
- 4

Tablica 2: Wyniki ekperymentu dla współczynnika mocy chłodzenia 'a'=1.2.

plik	czas wykonywania algorytmu[s]	l.p.	znalezione rozwiązanie	czas wykrycia [s]	błąd względny[%]
ftv47.atsp	120	1	1901	96	7.0
		2	1858	38	4.6
		3	1888	91	6.3
		4	1943	81	9.4
		5	1860	28	4.7
		6	1849	1	4.1
		7	1860	5	4.7
		8	1868	61	5.2
		9	1855	51	4.4
		10	1845	12	3.9
ftv170.atsp	240	1	3044	24	10.5
		2	3071	130	11.5
		3	3009	26	9.2
		4	3044	152	10.5
		5	2972	135	7.9
		6	3193	200	15.9
		7	2971	141	7.8
		8	3064	132	11.2
		9	3043	143	10.5
		10	3128	130	13.5
rbg403.atsp	360	1	2491	236	1.1
		2	2493	173	1.1
		3	2533	173	2.8
		4	2498	138	1.3
		5	2488	79	0.9
		6	2497	140	1.3
		7	2523	252	2.4
		8	2492	281	1.1
		9	2491	194	1.1
		10	2490	61	1.0

Tablica 3: Wyniki ekperymentu dla współczynnika mocy chłodzenia 'a'=1.6.

plik	czas wykonywania algorytmu[s]	l.p.	znalezione rozwiązanie	czas wykrycia [s]	błąd względny[%]
ftv47.atsp	120	1	1912	1	7.7
		2	1856	49	4.5
		3	1888	1	6.3
		4	1950	30	9.8
		5	1860	54	4.7
		6	1851	70	4.2
		7	1874	41	5.5
		8	1868	55	5.2
		9	1858	1	4.6
		10	1834	54	3.3
ftv170.atsp	240	1	3081	172	11.8
		2	3137	16	13.9
		3	3118	19	13.2
		4	3073	18	11.5
		5	3048	181	10.6
		6	3060	160	11.1
		7	3063	175	11.2
		8	3118	179	13.2
		9	3054	180	10.9
		10	3140	36	14.0
rbg403.atsp	360	1	2492	128	1.1
		2	2496	47	1.3
		3	2526	123	2.5
		4	2493	105	1.1
		5	2496	103	1.3
		6	2498	268	1.3
		7	2524	120	2.4
		8	2491	56	1.1
		9	2490	274	1.0
		10	2490	197	1.0

Tablica 4: Wyniki ekperymentu dla współczynnika mocy chłodzenia 'a'=1.9.

plik	czas wykonywania algorytmu[s]	l.p.	znalezione rozwiązanie	czas wykrycia [s]	błąd względny[%]
ftv47.atsp	120	1	1919	100	8.1
		2	1878	41	5.7
		3	1914	79	7.8
		4	1958	44	10.2
		5	1913	86	7.7
		6	1849	25	4.1
		7	1860	50	4.7
		8	1861	1	4.8
		9	1866	3	5.1
		10	1828	39	2.9
ftv170.atsp	240	1	3080	200	11.8
		2	3119	20	13.2
		3	3060	90	11.1
		4	3225	37	17.1
		5	3199	93	16.1
		6	3230	129	17.2
		7	3006	81	9.1
		8	3182	36	15.5
		9	3282	152	19.1
		10	3172	18	15.1
rbg403.atsp	360	1	2493	104	1.1
		2	2494	229	1.2
		3	2532	204	2.7
		4	2492	161	1.1
		5	2490	199	1.0
		6	2497	110	1.3
		7	2522	289	2.3
		8	2491	202	1.1
		9	2487	251	0.9
		10	2492	70	1.1

Wyniki uśrednione dla poszczególnych uruchomień zaprezentowane są w tabelach:

- 5
- 6
- 7

Tablica 5: Uśrednione wyniki eksperymentu dla współczynnika mocy chłodzenia 'a'=1.2.

plik	czas wykonywania algorytmu[s]	uśrednione rozwiązanie	błąd względny[%]	oczekiwane	uśredniony czas[s]
ftv47.atsp	120	1872.7	5.4	1776	46
ftv170.atsp	240	3053.9	10.8	2755	121
rbg403.atsp	360	2499.6	1.4	2465	173



Tablica 6: Uśrednione wyniki eksperymentu dla współczynnika mocy chłodzenia 'a'=1.6.

plik	czas wykonywania algorytmu[s]	uśrednione rozwiązanie	błąd względny[%]	oczekiwane	uśredniony czas[s]
ftv47.atsp	120	1875.1	5.6	1776	36
ftv170.atsp	240	3089.2	12.1	2755	114
rbg403.atsp	360	2499.6	1.4	2465	142

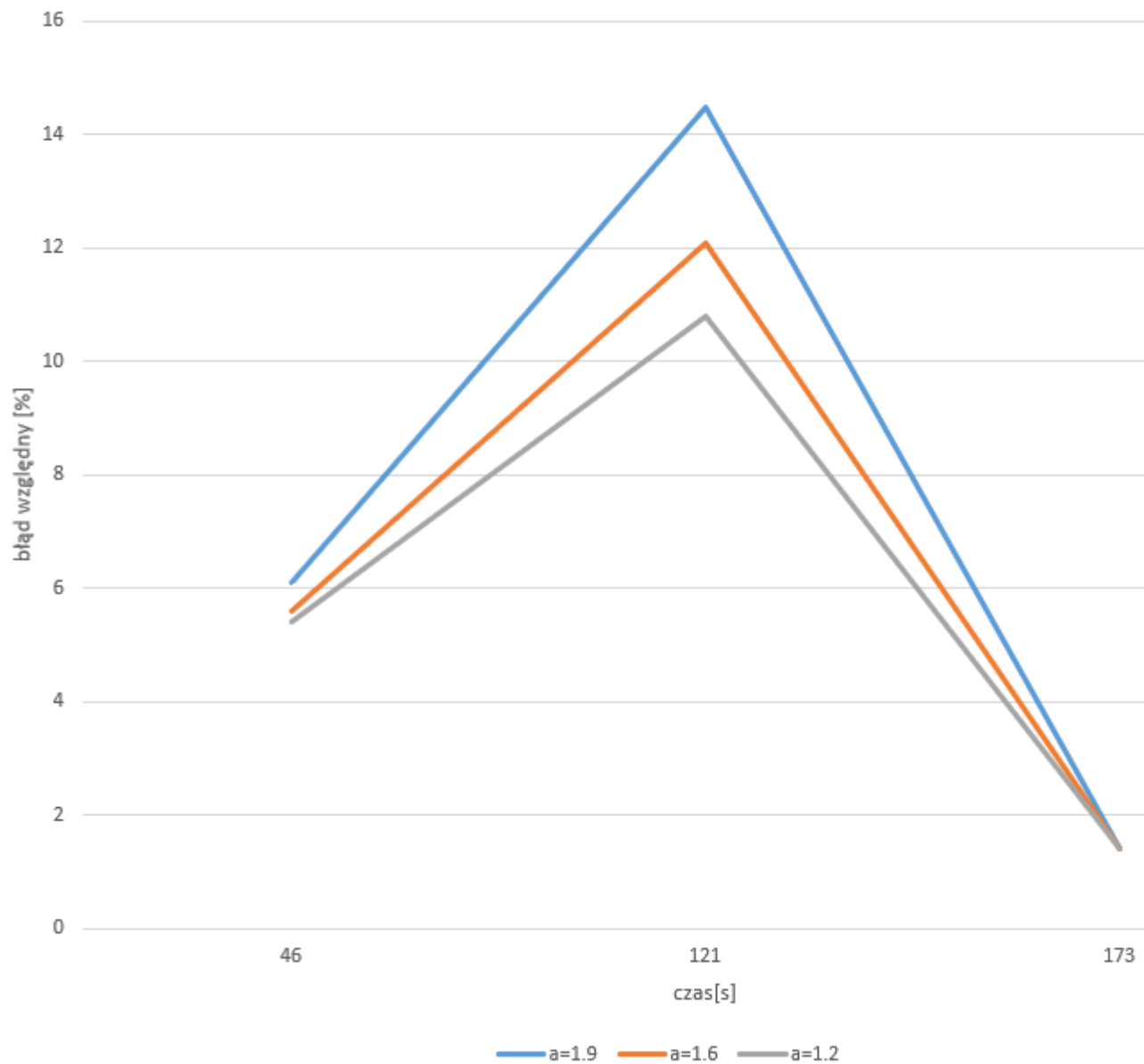
Tablica 7: Uśrednione wyniki eksperymentu dla współczynnika mocy chłodzenia 'a'=1.9.

plik	czas wykonywania algorytmu[s]	uśrednione rozwiązanie	błąd względny[%]	oczekiwane	uśredniony czas[s]
ftv47.atsp	120	1884.6	6.1	1776	47
ftv170.atsp	240	3155.5	14.5	2755	86
rbg403.atsp	360	2499	1.4	2465	182

## 4.2 Wykres

Na podstawie danych z tabel: 5, 6, 7 (na podstawie wartości z kolumn czasu średniego oraz uśrednionego błędu względnego) stworzono wykres jak na (Rys. 1).

Wykres błędu względnego w funkcji czasu dla uśrednionych wyników dla każdego pliku dla 3 różnych wartości współczynnika temperatury



Rysunek 1: Wykres błędu względnego w funkcji czasu dla uśrednionych wyników dla każdego pliku dla 3 różnych wartości współczynnika temperatury.

### 4.3 Najlepsze rozwiązania

Najlepsze znalezione ścieżki wraz z ich kosztami.

#### 4.3.1 ftv47.atsp

Koszt: 1828

Ścieżka:

0 – 37 – 38 – 18 – 17 – 13 – 34 – 46 – 36 – 35 – 14 – 23 – 12 – 32 – 7 – 31 – 30 –  
 5 – 24 – 4 – 29 – 3 – 6 – 10 – 11 – 8 – 9 – 33 – 27 – 28 – 2 – 41 – 43 – 42 – 22  
 – 19 – 44 – 15 – 16 – 45 – 39 – 21 – 40 – 47 – 26 – 1 – 25 – 20 – 0

#### 4.3.2 ftv170.atasp

Koszt: 2971

Ścieżka:

0 – 72 – 73 – 77 – 1 – 2 – 3 – 4 – 5 – 133 – 169 – 111 – 112 – 115 – 116 – 117 –  
 118 – 119 – 120 – 102 – 103 – 163 – 98 – 95 – 92 – 93 – 166 – 108 – 83 – 84 – 69  
 – 67 – 63 – 64 – 56 – 57 – 62 – 61 – 66 – 65 – 88 – 153 – 154 – 89 – 90 – 91 –  
 94 – 96 – 97 – 165 – 104 – 114 – 113 – 164 – 127 – 126 – 125 – 146 – 145 – 144 –  
 143 – 147 – 148 – 149 – 161 – 14 – 15 – 159 – 16 – 29 – 26 – 27 – 28 – 30 – 31  
 – 32 – 158 – 36 – 37 – 38 – 39 – 40 – 156 – 34 – 35 – 157 – 33 – 155 – 41 – 42 –  
 45 – 44 – 46 – 47 – 48 – 51 – 52 – 53 – 43 – 55 – 54 – 58 – 59 – 60 – 68 – 167  
 – 70 – 87 – 86 – 85 – 71 – 50 – 49 – 170 – 168 – 78 – 82 – 79 – 80 – 81 – 110 –  
 109 – 107 – 106 – 105 – 99 – 100 – 101 – 123 – 162 – 122 – 121 – 124 – 129 – 136  
 – 137 – 128 – 130 – 135 – 138 – 139 – 140 – 141 – 6 – 7 – 8 – 9 – 10 – 76 – 74  
 – 75 – 11 – 12 – 13 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 150 – 160 –  
 151 – 152 – 142 – 134 – 131 – 132 – 0

#### 4.3.3 rbg403.atasp

Koszt: 2487

Ścieżka:

0 – 92 – 347 – 352 – 361 – 51 – 259 – 184 – 265 – 326 – 390 – 207 – 45 – 340 – 231  
 – 205 – 204 – 222 – 250 – 388 – 348 – 329 – 294 – 272 – 356 – 391 – 210 – 182 –  
 149 – 146 – 381 – 183 – 17 – 12 – 242 – 70 – 295 – 201 – 89 – 316 – 120 – 116 –  
 366 – 321 – 289 – 236 – 227 – 320 – 264 – 105 – 19 – 57 – 64 – 15 – 245 – 248 –  
 235 – 121 – 169 – 334 – 385 – 67 – 189 – 88 – 96 – 66 – 273 – 395 – 373 – 30 –  
 302 – 118 – 252 – 371 – 83 – 230 – 226 – 325 – 111 – 369 – 237 – 135 – 270 – 171  
 – 63 – 18 – 97 – 276 – 49 – 296 – 133 – 176 – 31 – 400 – 221 – 199 – 136 – 142  
 – 301 – 351 – 293 – 81 – 22 – 21 – 181 – 346 – 94 – 314 – 313 – 323 – 168 – 213  
 – 28 – 318 – 179 – 368 – 193 – 104 – 349 – 344 – 178 – 100 – 44 – 24 – 13 – 288  
 – 254 – 223 – 309 – 164 – 25 – 202 – 125 – 74 – 354 – 342 – 198 – 128 – 186 – 53  
 – 305 – 215 – 6 – 123 – 399 – 266 – 90 – 72 – 122 – 119 – 115 – 114 – 62 – 345  
 – 173 – 319 – 78 – 382 – 144 – 159 – 7 – 177 – 387 – 34 – 174 – 185 – 232 – 244  
 – 80 – 131 – 355 – 73 – 172 – 26 – 32 – 274 – 188 – 299 – 298 – 291 – 379 – 217  
 – 203 – 153 – 143 – 271 – 370 – 322 – 93 – 180 – 359 – 166 – 300 – 206 – 267 –  
 317 – 234 – 228 – 398 – 396 – 327 – 262 – 261 – 68 – 335 – 10 – 402 – 287 – 281  
 – 246 – 163 – 3 – 42 – 360 – 353 – 341 – 151 – 11 – 397 – 35 – 124 – 269 – 211 –  
 233 – 324 – 212 – 38 – 86 – 59 – 157 – 1 – 87 – 137 – 307 – 75 – 79 – 69 – 304  
 – 5 – 56 – 150 – 145 – 190 – 138 – 224 – 77 – 156 – 258 – 372 – 278 – 297 – 127  
 – 192 – 338 – 82 – 247 – 308 – 394 – 393 – 99 – 389 – 110 – 275 – 154 – 238 –  
 229 – 14 – 375 – 191 – 187 – 141 – 392 – 312 – 194 – 380 – 362 – 241 – 113 – 126  
 – 55 – 333 – 158 – 339 – 195 – 280 – 279 – 106 – 165 – 285 – 282 – 357 – 328 –  
 257 – 109 – 134 – 243 – 208 – 162 – 48 – 47 – 112 – 91 – 167 – 54 – 108 – 256 –  
 249 – 160 – 365 – 170 – 284 – 139 – 4 – 239 – 117 – 95 – 268 – 16 – 367 – 310 –  
 52 – 40 – 263 – 50 – 220 – 197 – 152 – 76 – 401 – 29 – 65 – 364 – 303 – 377 –  
 311 – 27 – 253 – 140 – 101 – 129 – 255 – 161 – 148 – 147 – 374 – 363 – 315 – 175  
 – 103 – 209 – 41 – 102 – 240 – 219 – 358 – 46 – 33 – 376 – 58 – 8 – 214 – 155 –  
 20 – 331 – 378 – 337 – 23 – 225 – 332 – 37 – 36 – 283 – 277 – 39 – 85 – 260 –

43 — 84 — 107 — 386 — 2 — 61 — 350 — 218 — 384 — 383 — 9 — 286 — 60 — 98 — 251 —  
330 — 343 — 130 — 132 — 306 — 216 — 200 — 71 — 336 — 292 — 290 — 196 — 0

## 5 Wnioski dotyczące otrzymanych wyników

Wykres 1 dobrze obrazuje przebieg algorytmu dla różnych schematów chłodzenia. Dla trzech mierzonych instancji posiadających 48, 171 oraz 404 elementy. Wykres pokazuje, że instancja średnia sprawiła najwięcej trudności, najlepiej wypadła natomiast instancja największa.

W tej wersji algorytmu symulowanego wyżarzania zastosowane zostało chłodzenie Cauchy'ego [4] dla trzech różnych wartości współczynnika  $a = 1.2$ ,  $a = 1.6$  oraz  $a = 1.9$ . Linia zaznaczona na szaro na wykresie odpowiada współczynnikowi  $a = 1.2$  oraz dobrze obrazuje najmniejszy błąd względny spośród wymienionych trzech współczynników. Dla kolejnego współczynnika  $a = 1.6$  widać - pomarańczowa linia na wykresie - ,że ma większy błąd względny niż szara oraz mniejszy niż niebieska. Największy średni błąd względny został wyznaczony natomiast dla współczynnika największego:  $a = 1.9$ .

## 6 Wnioski

Algorytm Symulowanego Wyżarzania dla asymetrycznego problemu komiwojażera pomaga przybliżyć rozwiązanie dla dużych instancji macierzy kosztów wejściowych w nieporównywalnie krótszym czasie niż algorytmy: BruteForce, Branch&Bound. W zależności od instancji wyniki przybliżeń są różne. Np. dla instancji wejściowej: macierzy kosztów o rozmiarze 170 przybliżenie było średnio około 10 razy gorsze niż w przypadku instancji o rozmiarze 403. Może to być spowodowane faktem, że postacie macierzy są zbudowane inaczej, a generator liczb pseudolosowych działa w pewien określony sposób. Korzystano z *std::uniform\_real\_distribution*.

Kolejnym istotnym elementem jest schemat schładzania oraz zrozumienie znaczenia poszczególnych 'pokręteł' - współczynników. Dla temperatury wyższej zakłada się większy margines błędu, dla niższej mniejsza szansa na pomyłkę - uwzględnia się zawsze wynik lepszy lub wynik gorszy z dużo mniejszym prawdopodobieństwem dla małych temperatur bliskich zeru. Przetrzymywanie algorytmu w temperaturze bliskiej 0 może prowadzić do znalezienia lepszych wyników, kosztem dłuższego wykonywania. Zaimplementowane zostały trzy sposoby schładzania dla 3 różnych współczynników 'a' (1.2, 1.6, 1.9). Dla badanych grafów współczynnik 'a' powinien mieć małą wartość, ponieważ dla dużych wartości temperatura spada za szybko, przez co algorytm nie wyznacza optymalnych wyników. Wsp. 'a' powinien być większy lub równy 1. Dla współczynników bliskich 1 wyniki będą bardziej optymalne, jednak wtedy dłużej trwa chłodzenie - należy przygotować się na dłuższy czas wykonywania algorytmu - doświadczalnie dopasować warunek stopu. Podczas wyznaczania rozwiązania należy uwzględnić również liczbę epok; liczbę mówiącą o tym ile iteracji należy wykonać dla zadanej temperatury, po czym temperatura jest zmieniana. Podczas wykonywania testów zauważono, że zmniejszając tą wartość algorytm szybciej dąży do oczekiwanego wyniku, jednak po przekroczeniu pewnej wartości wynik niechętnie chce zmieniać stan, otrzymujemy wynik mniej optymalny, lecz w krótszym czasie.

Ponieważ implementowane schematy mają charakter pseudolosowy, teoretycznie mogłoby się zdarzyć, że wynik bliski oczekiwanemu mógłby się pojawić np. w pierwszych sekundach pracy algorytmu.

Ewentualne ulepszanie algorytmu polegałoby na zaimplementowaniu modułu machine-learningowego, którego zadaniem byłoby usprawnienie miejsca, w którym algorytm 'zamrażał' się i już nie znajdował dalszego rozwiązania. Np. w (Tab 3) widać, że sporo znalezionych rozwiązań zostało wykrytych we wczesnych fazach pracy algorytmu, a przez pozostały czas nie mogły się wydostać.

## 7 Literatura

- [1] Heurystyka (informatyka) - Wikipedia Wolna Encyklopedia, [data dostępu] 12.20.2020  
[https://pl.wikipedia.org/wiki/Heurystyka\\_\(informatyka\)](https://pl.wikipedia.org/wiki/Heurystyka_(informatyka))

- [2] 'List-Based Simulated Annealing Algorithm for Traveling Salesman Problem', Shi-hua Zhan,<sup>1,2</sup> Juan Lin,<sup>1</sup> Ze-jun Zhang,<sup>1</sup> and Yi-wen Zhong <sup>1,2</sup>, Published 13 Mar 2016, [data dostepu] 12.20.2020 <https://www.hindawi.com/journals/cin/2016/1712630/?fbclid=IwAR1VSkxS2jf4hpQzK6KxZ6stBBihojJeE5jweNOKmkQt>
- [3] 'A SYSTEMATIC PROCEDURE FOR SETTING PARAMETERS IN SIMULATED ANNEALING ALGORITHMS', Moon-Won Park\* and Yeong-Dae Kim<sup>O</sup> Department of Industrial Engineering, Korea Advanced Institute of Science and Technology (KAIST), 373-1 & sung-dong, Yusong-gu Daejeon, 305-701 Korea [data dostepu] 20.12.2020 <https://app.cs.amherst.edu/ccmcgeoch/cs34/papers/parkkimsdarticle-5.pdf>
- [4] 'Fast Simulated Annealing with a Multivariate Cauchy Distribution and the Configuration's Initial Temperature', Journal of the Korean Physical Society, Vol. 66, No. 10, May 2015, pp. 1457-1466, Fast Simulated Annealing with a Multivariate Cauchy Distribution and the Configuration's Initial Temperature [data dostepu] 21.12.2020 [https://www.researchgate.net/publication/279241787\\_Fast\\_simulated\\_annealing\\_with\\_a\\_multivariate\\_Cauchy\\_distribution](https://www.researchgate.net/publication/279241787_Fast_simulated_annealing_with_a_multivariate_Cauchy_distribution)